Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Sensor Failure Tolerant Supervisory Control

Kurt R. Rohloff

Abstract— **This paper discusses problems related to partial observation supervisory controllers with possibly faulty sensors using the framework of discrete-event systems. At initialization all sensors are operational such that the sensors observe occurrences of events and transmit those observations to the controller, but, when a sensor fails, it ceases to send signals to the controller. A new version of observability is introduced that is part of the necessary and sufficient conditions for controller existence under the assumption of faulty sensors. A polynomial-time construction is given that can be used to test for and then synthesize a non-blocking controller with faulty sensors using standard supervisory control methods.**

I. Introduction

When designing a controller for a system to match a given specification it is generally desirable in safety-critical applications for the controller to be fault tolerant. That is, it is desirable to design controllers in a redundant manner such that even if the controller fails partially, it will still be able to achieve its desired task, or at least not fail catastrophically. This field, called Fault Tolerant Control (FTC), has been a very active in many control theoretic research areas, including discrete-event systems ([2]).

In the standard partial observation supervisory control model as discussed in [3], controllers have sensors to observe occurrences of a subset of system events. Supervisory controllers are usually designed with the assumption that the controllers are fault-free. However, this assumption of controller infallibility may not be reasonable over the full life-cycle of a control system due to the natural deterioration of a controller over time. For instance, control circuitry may degenerate as a control system ages, a control actuator may become stuck, or sensors may fail.

A controller's sensors are normally assumed to be deterministic in that sensors always communicate event occurrences to the controller. However, as was indicated above, sensors failures may occur such that individual sensors may cease to send correct signals to a controller. This paper focuses on problems related to the testing of controller existence and performing controller synthesis for a given specification when the controller's sensors may fail. It is assumed in this paper that sensors fail permanently such that before failure a sensor operates normally as in [3], but after failure no signals are sent from the sensor to the controller such that, the previously observable event becomes effectively unobservable after failure. It is also assumed that not only may the sensors for the control systems fail, but the sensor failures are sufficiently uncommon such that no two sensors may both be failed at any given time. As expected, sensor failures cannot be directly observed.

The paper is structured as follows. The next section of this paper presents preliminary definitions and notation from supervisory control. Section III presents an observability property for systems with faulty sensors and discusses existence properties for fault-tolerant control systems with unreliable sensors. Section IV presents a method for testing the existence of sensor-failure tolerant controllers to satisfy a specification. Section V discusses methods for the synthesis of sensor-failure tolerant control systems based on known methods in the supervisory controls literature. Section VI closes the paper with a review of the results contained herein. for the sake of brevity, the proofs of the lemmas and theorems in this paper are shown in [5].

II. Preliminaries and Notation

To aid the reader, this section gives a review of necessary concepts of partial-observation supervisory control as presented in [3]. Due to the necessary brevity of this paper, a deeper introduction can be found in [1]. In the supervisory control framework, system and specification behaviors are modelled as languages of the automata $G = (X^G, x_0^G, \Sigma^G, \delta^G, X_m^G)$ and $H = (X^H, x_0^H, \Sigma^H, \delta^H, X_m^H)$, respectively, where X^G and X^H are sets of states, x_o^G and x_o^H are initial states, $\Sigma^H = \Sigma^G$ is the common event set of the automata, $\delta^G : X^G \times \Sigma^G \to X^G$ and $\delta^H : X^H \times \Sigma^G \to X^H$ are the (possibly partial) state transition functions, and X_m^G and X_m^H are the marked states of G and H, respectively.

Following the formalisms of [3], controllers may have a set of sensors to observe a set of system events $\Sigma_o \subseteq \Sigma^G$ with each sensor assigned to deterministically observe all occurrences of exactly one event. Furthermore, on the occurrence of observable events, controllers may be given sufficient actuation to selectively disable a subset of the controllable events $\Sigma_c \subseteq \Sigma^G$. Controllers can be realized as finite state automata that observe some events and control a potentially different set of events. Controllers should not be able to disable uncontrollable events and control actions should not update on the occurrence of unobservable events. The set of unobservable events, $\Sigma_{uo} = \Sigma^G \setminus \Sigma_o$, is the set of events whose occurrence can never be observed even in the absence of sensor failures. Similarly, the set of uncontrollable events, $\Sigma_{uc} = \Sigma^G \setminus \Sigma_c$ are the events whose occurrence cannot be regulated. Given a controller S and a system G, the composed system of S controlling G with perfect sensors is denoted as the controlled system S/G.

This research was supported by the NSF grant CCR 00-85917 ITR.

K. Rohloff is with the Coordinated Science Laboratory at The University of Illinois, 1308 West Main St., Urbana, IL 61801, USA. krohloff@control.csl.uiuc.edu

0-7803-9568-9/05/$20.00 ©2005 IEEE

For a given set of observable events $\Sigma_o \subseteq \Sigma^G$, a natural projection operation $P : \Sigma^G \to \Sigma_o$ is used to model a controller's observations of system behavior. For the empty event ϵ, $P(\epsilon) = \epsilon$, and for a string of events s and an event σ,

$$P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{otherwise} \end{cases} .$$

There is also a corresponding inverse projection operation $P^{-1} : {\Sigma^{G}}^{*}_{o} \to 2^{\Sigma^{G^*}}$.

Three important properties related to controller existence are controllability and observability and M-closure.

Definition 1: [4] Consider the languages K and M such that $M = \overline{M}$ and the set of uncontrollable events Σ_{uc}. The language K is *controllable* with respect to M and Σ_{uc} if

$$\overline{K}\Sigma_{uc} \cap M \subseteq \overline{K}. \tag{1}$$

Definition 2: [3] Consider the sets of languages K and M such that $M = \overline{M}$ and the set controllable, Σ_c, and observable Σ_o events. The language K is *observable* with respect to M, $P(\cdot)$ and Σ_c if for all $t \in \overline{K}$ and for all $\sigma \in \Sigma_c$,

$$\begin{aligned} &\left[\left(t\sigma \notin \overline{K}\right) \wedge (t\sigma \in M)\right] \Rightarrow \\ &\quad \left[\left(P^{-1}\left[P(t)\right]\sigma \cap \overline{K} = \emptyset\right) \wedge (\sigma \in \Sigma_c)\right]. \end{aligned} \tag{2}$$

Definition 3: Consider the languages K and M. The set K is *M-closed* if $K = \overline{K} \cap M$.

The above definitions of controllability, observability and M-closure are central in the following controller existence theorem called the controllability and observability theorem.

Theorem 1: [3] For a finite state automaton system G, a finite state automaton specification H such that $\mathcal{L}_m(H) \subseteq \mathcal{L}_m(G)$, a set of controllable events Σ_c and a set of observable events Σ_o, there exists a nonblocking partial observation controller S such that $\mathcal{L}_m(S/G) = \mathcal{L}_m(H)$ and $\mathcal{L}(S/G) = \overline{\mathcal{L}_m(H)}$ if and only if the following conditions hold:

1) $\mathcal{L}_m(H)$ is controllable w.r.t. $\mathcal{L}(G)$ and Σ_{uc}.
2) $\mathcal{L}_m(H)$ is observable w.r.t. $\mathcal{L}(G)$, Σ_o and Σ_c.
3) $\mathcal{L}_m(H)$ is $\mathcal{L}_m(G)$-closed.

One might think that due to Theorem 1 that for there to exist a supervisory control system that is tolerant to single sensor failures, one could ensure that for all $\sigma \in \Sigma_o$ $\mathcal{L}_m(H)$ is observable with respect to $\mathcal{L}(G)$, $\Sigma_o \setminus \{\sigma\}$ and Σ_c. That is, one might expect that if any one event $\sigma \in \Sigma_o$ is made unobservable during control operation, but the specification $\mathcal{L}_m(H)$ is always observable with respect to any $\Sigma^G \setminus \{\sigma\}$ for any $\sigma \in \Sigma_o$, then there would exist a nonblocking controller S tolerant to sensor failures such that the controller behavior matches $\mathcal{L}_m(H)$. Unfortunately, this is not the case. Consider the following example.

Example 1: Consider the system automaton G and the specification automaton H seen in Figure 1.

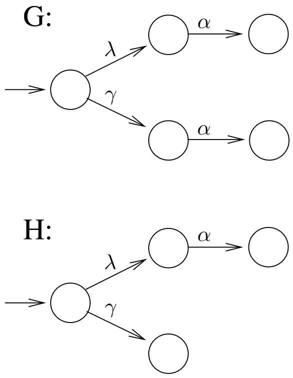

Fig. 1. The system automaton G and the specification automaton H for Example 1.

Let $\Sigma_c = \{\alpha\}$. If $\Sigma_o = \{\lambda\}$, the proper control action at initialization would be to disable α at initialization. Similarly, if $\Sigma_o = \{\gamma\}$, then the proper control action would be to enable α at initialization. However, if $\Sigma_o = \{\gamma, \lambda\}$, and if either the sensor for γ or λ may fail at initialization, and the controller can have no direct observations of sensor failure, then there is no correct initial control action. Therefore, it is not possible to synthesize a controller with possibly faulty sensors for this example when $\Sigma_o = \{\gamma, \lambda\}$ to match the specification $\mathcal{L}(H)$ even though for all $\sigma \in \Sigma_o$, $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, $\Sigma_o \setminus \{\sigma\}$ and Σ_c.

III. Observability With Respect to Sensor Failures

As was discussed in the introduction, it is assumed that a controller's sensors fail in such a way that after failure a sensor halts sending signals to the controller. Furthermore, it is assumed that only one controller can have failed at any given time. With this scenario, an interesting problem is to decide if there exists a controller S for a system G, a specification H, controllable events Σ_c and sensors for observable events Σ_o that can fail as described above such that when the controller is coupled with the system, the specification can be satisfied even if a sensor may fail.

With this motivation the deterministic project operation (originally $P : \Sigma^{G^*} \to \Sigma_o^*$) is generalized to the faulty sensor projection operation $P^f : \Sigma^{G^*} \to 2^{\cup_{\sigma \in \Sigma_o}\Sigma_o^*\{\Sigma_o \setminus \sigma\}^*}$ where for a string $s \in \Sigma^{G^*}$, the set $P^f(s)$ is all strings that could be observed for a system with faulty sensors as described above. Let projection operation $P_\sigma : \Sigma^{G^*} \to (\Sigma_o \setminus \{\sigma\})^*$ be the same as the $P(\cdot)$ operation except that events in $\Sigma_o \setminus \{\sigma\}$ are retained in the projection instead of events in Σ_o. Then,

$$P^f(s) = \{P(s_1)P_\sigma(s_2) | s_1 s_2 = s, \sigma \in \Sigma_o\} \tag{3}$$

Therefore, if a string $s \in \Sigma^{G^*}$ occurs in the system such that the sensor for the event σ fails after the occurrence of s_1 where $s_1 s_2 = s$, then $P(s_1)$ is observed before the sensor failure and $P_\sigma(s_2)$ is observed after the sensor failure. Consequently, $P(s_1)P_\sigma(s_2)$ is the string observed by the controller resulting from the sensor failure as described, and $P^f(s)$ is the set of all strings that could be observed due to s and the failure of an event in Σ_o.

As with the original projection operation the faulty sensor projection operation also has an inverse projection operation $P^{f^{-1}} : \cup_{\sigma \in \Sigma_o}\Sigma_o^*\{\Sigma_o \setminus \sigma\}^* \to 2^{\Sigma^{G^*}}$. Formally, $P^{f^{-1}}(s) = \{s' | s \in P^f(s')\}$. The inverse operation $P^{f^{-1}}(s)$ denotes all strings of behavior that could generate the observation of s. Note that because the faulty sensor projection $P^f(s)$ denotes the set of observed strings which could be generated by s, then ${P^f}^{-1}(P^f(s))$ denotes all strings that *might* generate an observed set of strings which could also be generated by s. Although it is not shown here, the $P^f(\cdot)$ and ${P^f}^{-1}(\cdot)$ operations preserve regularity.

Due to the insufficiency of observability as a necessary and sufficient condition for fault-tolerant controller existence, a new property is now introduced called *observability with respect to sensor failure*, or *sensor failure observability* for short.

Definition 4: Consider the languages K and M such that $M = \overline{M}$, the set of controllable events, Σ_c and observable events Σ_o. The language K is *observable with respect to sensor failure* with respect to M, $P^f(\cdot)$ and Σ_c if for all $t \in \overline{K}$ and for all $\sigma \in \Sigma_c$,

$$\begin{aligned}&\left[\left(t\sigma \notin \overline{K}\right) \wedge (t\sigma \in M)\right] \Rightarrow \qquad (4)\\ &\quad \left[\left({P^f}^{-1}\left[P^f(t)\right]\sigma \cap \overline{K} = \emptyset\right) \wedge (\sigma \in \Sigma_c)\right].\end{aligned}$$

It is shown in Section V that sensor failure observability is part of the set of necessary and sufficient faulty-sensor controller existence conditions, similar to the observability property in Theorem 1. Although it is not shown here, sensor failure observability is closed with respect to language intersection operations, but not with respect to language union operations. Some important properties of sensor failure observability are now shown.

A. Properties of Sensor Failure Observability

Due to Example 1, if K is observable with respect to M, $\Sigma_o \setminus \{\sigma\}$ and Σ_c for all $\sigma \in \Sigma_o$, it is not necessary for K to be sensor failure observable with respect to M, Σ_o and Σ_c. However, the reverse is true as demonstrated in the Proposition 1 below. First, some preliminary lemmas are shown without their proofs for reasons of brevity.

Lemma 1: For the $P_\sigma(\cdot)$ and $P^f(\cdot)$ functions as defined above, for any $t \in \Sigma^{G^*}$, $P_\sigma(t) \in P^f(t)$.

Lemma 2: For the $P_\sigma^{-1}(\cdot)$ and ${P^f}^{-1}(\cdot)$ functions as defined above, for all $\sigma \in \Sigma_o$ and any $t \in \Sigma^{G^*}$, $P_\sigma^{-1}(L) \subseteq {P^f}^{-1}(L)$.

Lemma 3: For the ${P^f}^{-1}(\cdot)$ function as defined above, for all $\sigma \in \Sigma_o$ and any languages $L \subseteq \Sigma^{G^*}$ and $L' \subseteq \Sigma^{G^*}$ such that $L \subseteq L'$, ${P^f}^{-1}(L) \subseteq {P^f}^{-1}(L')$.

Proposition 1: If K is sensor failure observable with respect to M, Σ_o and Σ_c, then $\forall \sigma \in \Sigma_o$, K is observable with respect to M, $\Sigma_o \setminus \{\sigma\}$ and Σ_c.

IV. Testing Sensor Failure Observability

A method to test sensor failure observability is now shown based on the construction of two deterministic automata $\vec{G}$ and $\vec{H}$ and two sets of events $\vec{\Sigma}_o$ and $\vec{\Sigma}_c$ from G, H, Σ_c and Σ_o in polynomial time such that $\mathcal{L}_m(\vec{H})$ is observable (in the sense of [3]) with respect to $\mathcal{L}(\vec{G})$, $\vec{\Sigma}_o$ and $\vec{\Sigma}_c$ if and only if $\mathcal{L}_m(H)$ is sensor failure observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c. Therefore, the standard methods for testing observability can be used with the $\vec{G}$ and $\vec{H}$ constructions to test sensor failure observability. To facilitate these constructions, two intermediate constructions of the automata, $\tilde{G}$ and $\tilde{H}$ from G and H are given such that $P^f(\mathcal{L}_m(G)) = P(\mathcal{L}_m(\tilde{G}))$ and $P^f(\mathcal{L}_m(H)) = P(\mathcal{L}_m(\tilde{H}))$.

The intuition behind the construction of $\tilde{G}$ is that if there are n observable events $\Sigma_o = \{\sigma_1, \ldots, \sigma_n\}$, then the system G has $n+1$ modes of operation with respect to sensor failure. In the initial mode of operation, mode 0, all sensors for observable events are operational. However, when the sensor for event $\sigma_i \in \{\sigma_1, \ldots, \sigma_n\}$ fails, the system then enters mode i where σ_i event occurrences are no longer observable. Note that the underlying state transition behavior in all modes of operation should be identical to the state transition behavior of G, but the observability properties of events occurrences are altered between various modes of operation.

With this in mind, for the observable event set $\Sigma_o = \{\sigma_1, \ldots, \sigma_n\}$, the automata $G_0, G_1, \ldots, G_n$ are used to model the system observation behavior in the various modes of operation with respect to sensor failure. The automaton G_i represents the behavior of the system in mode i.

For $i \in \{0, \ldots, n\}$ the automaton $G_i = (X_i^G, \Sigma_i, \delta_i^G, X_{mi}^G)$ is a copy of the original system automaton G with the relabelling of the states and some events. The state set X_i^G of G_i is a copy of X^G such that for every $x \in X^G$ from G there is a corresponding state $x_i \in X_i^G$. A one-to-one function $\Phi_i : X^G \rightarrow X_i^G$ is defined which translates a state in X^G to its corresponding state in X_i^G such that $\Phi_i(x) = x_i$. The inverse operation $\Phi_i^{-1}(\cdot)$ is defined in the usual manner. Similarly, all marked states $x_{mi} \in X_{mi}^G$ are copies of marked states $x_m \in X_m^G$ according to the $\Phi_i(\cdot)$ function.

As noted above, in the initial mode of operation, all sensors are operational. Therefore the state transition structure of G_0 is identical to G such that $\delta_0^G(x_0, \sigma) = \Phi_0(\delta^G(x, \sigma))$.

However, for the other n modes of operation, each mode corresponds to the failure of an event sensor. There are therefore n new unobservable events $\Sigma_o^f = \{\sigma_1^f, \ldots, \sigma_n^f\}$ such that every transitions originally labelled by a σ_i event in G is replaced in G_i by a transition labelled by the corresponding unobservable event σ_i^f. An occurrence of a σ_i^f event in mode i signifies that σ_i occurred in the system but this event is not observed due to sensor failure.

Therefore, the set of events $\Sigma_i = \{\sigma_1, \ldots, \sigma_i^f, \ldots, \sigma_n\}$ is a copy of Σ^G with σ_i replaced by σ_i^f. A one-to-one function $\Psi_i : \Sigma^G \rightarrow \Sigma_i$ is defined to translate the events in Σ^G to the corresponding events in Σ_i such that $\Psi_i(\sigma) = \sigma_i^f$ if $\sigma = \sigma_i$ and $\Psi_i(\sigma) = \sigma$ otherwise. A corresponding inverse operation $\Psi_i^{-1}(\cdot)$ is defined in the usual manner. The state transition structure of G_i, $\delta_i^G : X_i^G \times \Sigma_i \rightarrow X_i^G$ is formally defined as $\delta_i^G(x_i, \sigma) = \Phi_i(\delta^G(x, \Psi_i^{-1}(\sigma)))$.

An example of the construction of the modes $G_0, \ldots, G_n$ can be seen in Figure 3 which are constructed from the automaton G seen in Figure 2. Note that $\Sigma_o = \{\alpha, \beta\}$, $\sigma_1 = \alpha$ and $\sigma_2 = \beta$.

Before $\tilde{G}$ is formally defined, its overall behavior is

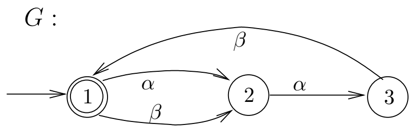

Fig. 2. The automaton G.

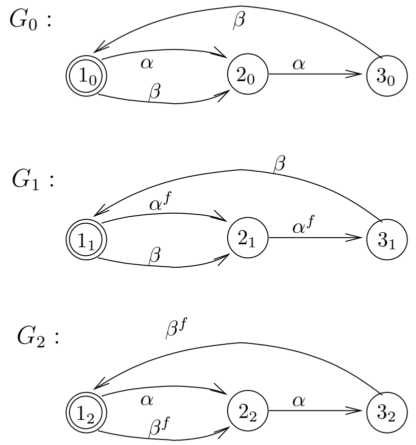

Fig. 3. Mode automata G_0, G_1 and G_2 constructed from G in Figure 2.

described. The mode automata $\{G_0, \ldots, G_n\}$ are used to construct the $\tilde{G}$ automaton through concatenation such that $\tilde{G}$ simulates the sensor signalling under the assumption of possible sensor failures and $P^f(\mathcal{L}_m(G)) = P(\mathcal{L}_m(\tilde{G}))$. Due to the sensor failure dynamics described in this paper, the automaton $\tilde{G}$ initially is in mode 0 and all observable events can be observed. On the failure of the sensor for event σ_i, the system enters mode i such that occurrences of σ_i are no longer observable. Hence, if $\tilde{G}$ is the concatenation of the various mode automata $\{G_0, \ldots, G_n\}$, then $\tilde{G}$ is initially in the mode modelled by G_0, but on the failure of the sensor for σ_i, $\tilde{G}$ enters the mode modelled by G_i. For G_i, σ_i^f events are used in place of σ_i events in order to model the change in event observability due to the mode switching. To govern the mode transition dynamics in $\tilde{G}$ a new set of events $F^{\Sigma_o} = \{f^{\sigma_1}, \ldots, f^{\sigma_n}\}$ is defined such that f^{σ_i} represents the failure of the sensor for event σ_i. Therefore, on the occurrence of event f^{σ_i}, $\tilde{G}$ should transition from mode 0 to mode i.

With this description of the overall behavior of $\tilde{G}$, this automata is defined formally as $(X^{\tilde{G}}, \Sigma^{\tilde{G}}, x_0^{\tilde{G}}, \delta^{\tilde{G}}, X_m^{\tilde{G}})$. For the state space of $\tilde{G}$, define $X^{\tilde{G}} = X_0^G \cup \cdots \cup X_n^G$ and $X_m^{\tilde{G}} = X_{0m}^G \cup \cdots \cup X_{nm}^G$. For the event set, let $\Sigma^{\tilde{G}} = \Sigma^G \cup \Sigma_o^f \cup F^{\Sigma_o}$. The system is initially in mode 0 so the initial state of $\tilde{G}$ is defined to be $\Phi_0(x_0^G)$ so that if no sensor failure occurs the set of behaviors that could be observed due to state transitions in $\tilde{G}$ equal the set of behaviors that could be observed due to state transitions in G. The state transition function $\delta^{\tilde{G}} : X^{\tilde{G}} \times \Sigma^{\tilde{G}} \to X^{\tilde{G}}$ is defined as follows:

$$\delta^{\tilde{G}}(x, \sigma) = \qquad (5)$$

$$\begin{cases} \delta^{G_i}(x, \sigma) & \text{if } \delta^{G_i}(x, \sigma)! \\ \Phi_i(\Phi_0^{-1}(x)) & \text{if } \left(x \in X^{G_0}\right) \wedge (\sigma = f^{\sigma_i}) \\ \text{undefined} & \text{otherwise} \end{cases} .$$

For the unary operator !, $f(\alpha)!$ is true if $f(\cdot)$ is defined for input α, false otherwise. Note that if G and H are deterministic, then $\tilde{G}$ and $\tilde{H}$ are also deterministic. Examples of the $\tilde{G}$ and $\tilde{H}$ constructions are now given.

Example 2: Consider the system automaton G in Figure 2 and the specification automaton H in Figure 4.

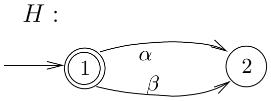

Fig. 4. The specification automaton H.

Using the method outlined above, the automaton $\tilde{G}$ constructed from G can be seen in Figure 5 and the automaton $\tilde{H}$ constructed from H can be seen in Figure 6.

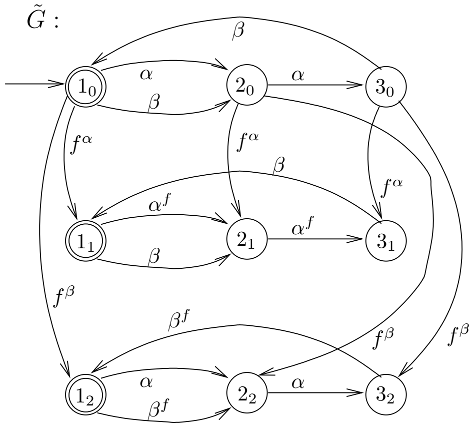

Fig. 5. The automaton $\tilde{G}$ constructed from G in Figure 1.

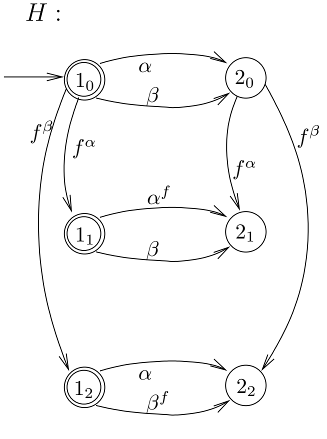

Fig. 6. The automaton $\tilde{H}$ constructed from H in Figure 4.

It is now shown that $P^f(\mathcal{L}_m(G)) = P(\mathcal{L}_m(\tilde{G}))$ if a slight abuse of notation is allowed to extend the definition of $P(\cdot)$ to $P : \Sigma^{\tilde{G}^*} \to \Sigma_o^*$. That is, the domain of $P(\cdot)$ is enlarged to be defined over $\Sigma^{\tilde{G}^*}$.

Theorem 2: Suppose an automaton G and an observable event set Σ_o are given. Then, for the corresponding $\tilde{G}$ construction as described above, $P^f(\mathcal{L}_m(G)) = P(\mathcal{L}_m(\tilde{G}))$.

Now that the $\tilde{G}$ construction has been presented, constructions for $\vec{G}$, $\vec{H}$, $\vec{\Sigma}_o$ and $\vec{\Sigma}_c$ are shown such that $\mathcal{L}_m(H)$ is sensor failure observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c if and only if $\mathcal{L}_m(\vec{H})$ is observable with respect to $\mathcal{L}(\vec{G})$, $\vec{\Sigma}_o$ and $\vec{\Sigma}_c$.

The construction of $\vec{G}$ is based on the $\tilde{G}$ automaton. The automaton $\vec{G}$ is formally defined as the 5-tuple $(X^{\vec{G}}, \Sigma^{\vec{G}}, x_0^{\vec{G}}, \delta^{\vec{G}}, X_m^{\vec{G}})$. Let $X^{\vec{G}} = X^{\tilde{G}} \cup \{d, d_m\}$ and $X_m^{\vec{G}} = X_m^{\tilde{G}} \cup \{d_m\}$. That is, the state spaces of $\tilde{G}$ and $\vec{G}$ are identical except that $\vec{G}$ has two additional states, one of which is marked. Also, $\Sigma^{\vec{G}} = \Sigma^{\tilde{G}}$ and $x_0^{\vec{G}} = x_0^{\tilde{G}}$ so that $\tilde{G}$ and $\vec{G}$ have identical event sets and initial states. The state transition function $\delta^{\vec{G}} : X^{\vec{G}} \times \Sigma^{\vec{G}} \to X^{\vec{G}}$ is defined as follows:

$$\delta^{\vec{G}}(x,\sigma) =$$
$$\begin{cases} \delta^{\tilde{G}}(x,\sigma) & \text{if } \delta^{\tilde{G}}(x,\sigma)! \\ d & \text{if } \left(\begin{array}{l} \left(x \in X^{G_i} | i \neq 0\right) \wedge (\sigma = \sigma_i) \wedge \\ \left(\nexists s \in \Sigma^{\vec{G}^*} | \delta^{\tilde{G}}(x, \sigma s) \in X_m^{\tilde{G}}\right) \end{array} \right) \\ d_m & \text{if } \left(\begin{array}{l} \left(x \in X^{G_i} | i \neq 0\right) \wedge (\sigma = \sigma_i) \wedge \\ \left(\exists s \in \Sigma^{\vec{G}^*} | \delta^{\tilde{G}}(x, \sigma s) \in X_m^{\tilde{G}}\right) \end{array} \right) \\ \text{undefined} & \text{otherwise} \end{cases} .$$

For a control system, the specification automaton H can be used to construct $\tilde{H}$ and $\vec{H}$, similar to how $\tilde{G}$ and $\vec{G}$ are constructed from G. Also define $\vec{\Sigma}_c = \Sigma_c \cup \{\sigma_i^f | \Phi_i^{-1}(\sigma_i^f) \in \Sigma_c\}$ and $\vec{\Sigma}_o = \Sigma_o$.

Examples of the $\tilde{G}$ and $\tilde{H}$ constructions are now given.

Example 3: Consider the system automaton G in Figure 2 and the specification automaton H in Figure 4. Furthermore, let $\Sigma_o = \{\alpha, \beta\}$ and let $\Sigma_c = \{\alpha, \beta\}$.

Using the method outlined above, the automaton $\vec{G}$ constructed from G can be seen in Figure 7 and the automaton $\vec{H}$ constructed from H can be seen in Figure 8. Furthermore, $\vec{\Sigma}_o = \{\alpha, \beta\}$ and $\vec{\Sigma}_c = \{\alpha, \beta, \alpha^f, \beta^f\}$.

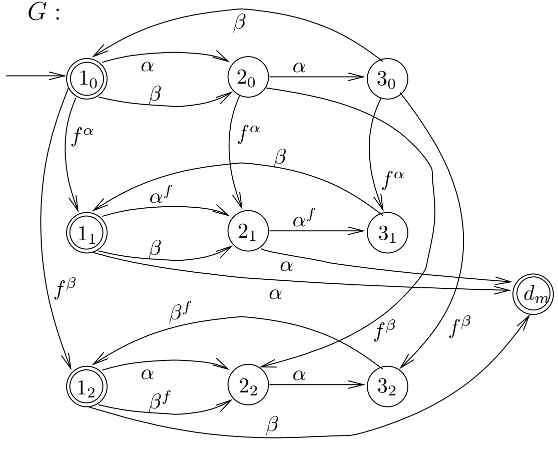

Fig. 7. The automaton $\vec{G}$ constructed from G in Figure 1.

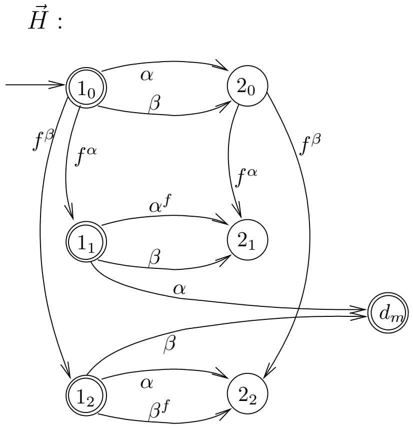

Fig. 8. The automaton $\vec{H}$ constructed from H in Figure 4.

For the intuition behind the constructions of $\vec{G}$ and $\vec{H}$, suppose there are two strings $t_1, t_2 \in \mathcal{L}(G) \cap \overline{\mathcal{L}_m(H)}$ and an event $\sigma \in \Sigma_c$ such that $t_1\sigma \in \overline{\mathcal{L}_m(H)}$, $t_2\sigma \in \mathcal{L}(G) \setminus \overline{\mathcal{L}_m(H)}$ and $P_f(t_1) \cap P_f(t_2) \neq \emptyset$. Because $P_f(t_1) \cap P_f(t_2) \neq \emptyset$, then it is possible for the sensors of the system to fail in such a way that the observation generated by t_1 and the observation generated by t_2 are identical. Therefore, there are two strings $\vec{t}_1, \vec{t}_2 \in \overline{\mathcal{L}_m(\vec{H})}$ such that $P(\vec{t}_1) = P(\vec{t}_2)$. However, if there is an event $\sigma \in \Sigma_c$ such that $t_1\sigma \in \overline{\mathcal{L}_m(H)}$ and $t_2\sigma \in \mathcal{L}(G) \setminus \overline{\mathcal{L}_m(H)}$, then it is possible that the correct control action after t_2 cannot be known. This is tested for in the $\vec{G}$ construction with the d and d_m transitions, so that if $t_1\sigma \in \overline{\mathcal{L}_m(H)}$, then $\vec{t}_1\sigma \in \overline{\mathcal{L}_m(\vec{H})}$, and if $t_2\sigma \in \mathcal{L}(G) \setminus \overline{\mathcal{L}_m(H)}$, then $\vec{t}_2\sigma \in \mathcal{L}(\vec{G}) \setminus \overline{\mathcal{L}_m(\vec{H})}$. This effectively converts the sensor failure observability test of $\mathcal{L}_m(H)$ into an observability test of $\mathcal{L}_m(\vec{H})$. This is formalized in the proof of Theorem 3.

Theorem 3: Suppose G, H, Σ_o and Σ_c are given and $\vec{G}$, $\vec{H}$, $\vec{\Sigma}_o$ and $\vec{\Sigma}_c$ are constructed from them. Then, $\mathcal{L}_m(H)$ is sensor failure observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c if and only if $\mathcal{L}_m(\vec{H})$ is observable with respect to $\mathcal{L}(\vec{G})$, $\vec{\Sigma}_o$ and $\vec{\Sigma}_c$.

Note that the sizes of the state spaces of $\vec{G}$ and $\vec{H}$ are in $O(|\Sigma_o||X^G|)$ and $O(|\Sigma_o||X^H|)$ respectively. Therefore, $\vec{G}$ and $\vec{H}$ can all be constructed in polynomial time with respect to the sizes of G, H, Σ^G. Now that the standard methods for testing observability can also be used to test sensor failure observability. It is well known that observability can be decided in polynomial time [6], so sensor failure observability can therefore be decided in polynomial time.

V. Controller Synthesis with Faulty Sensors

A controller $S : \Sigma_o^* \to 2^{\Sigma_c} \cup \Sigma_{uc}$ is considered a map from observed strings in Σ_o^* to a set of enabled events in Σ^G. Therefore, if t is observed by controller S, then $S(t)$ is the set of events enabled by S. The composed system of S controlling G under the assumption of faulty sensors is denoted as $S\phi G$. The language generated by $S\phi G$, denoted by $\mathcal{L}(S\phi G)$, is defined recursively as follows:

- $\epsilon \in \mathcal{L}(S\phi G)$.

- $s \in \mathcal{L}(S\phi G)$, $s\sigma \in \mathcal{L}(G)$, and $\exists t \in P^f(s)$ such that $\sigma \in S(t)$ if and only if $s\sigma \in \mathcal{L}(S\phi G)$.

The language marked by $S\phi G$, denoted by $\mathcal{L}_m(S\phi G)$, is $\mathcal{L}(S\phi G)\cap\mathcal{L}_m(G)$. Note that if $s \in \mathcal{L}(S\phi G)$, $s\sigma \in \mathcal{L}(G)$, and $\exists t, t' \in P^f(s)$ such that $\sigma \in S(t)$ and $\sigma \notin S(t')$, then $s\sigma \in \mathcal{L}(S\phi G)$. The concept of sensor failure observability is part of the set of necessary and sufficient faulty-sensor controller existence conditions, similar to observability in Theorem 1.

Theorem 4: For a finite state automaton system G, a finite state automaton specification H such that $\mathcal{L}_m(H) \subseteq \mathcal{L}(G)$, a set of controllable events Σ_c and a set of observable events Σ_o with sensors that may fail as described above, there exists a nonblocking partial observation faulty sensor controller S such that $\mathcal{L}_m(S\phi G) = \mathcal{L}_m(H)$ and $\mathcal{L}(S\phi G) = \overline{\mathcal{L}_m(H)}$ if and only if the following conditions hold:

1) $\mathcal{L}_m(H)$ is controllable with respect to $\mathcal{L}(G)$ and Σ_{uc}.
2) $\mathcal{L}_m(H)$ is sensor failure observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c.
3) $\mathcal{L}_m(H)$ is $\mathcal{L}_m(G)$-closed.

Note that if $s \in \mathcal{L}(S\phi G)$, $s\sigma \in \overline{\mathcal{L}_m(H)}$, and $\exists t, t' \in P^f(s)$ it is possible that $\sigma \in S(t)$ and $\sigma \notin S(t')$ and still have $s\sigma \in \mathcal{L}(S\phi G)$. However, the concept of sensor-failure observability guarantees that if $s\sigma \in \overline{\mathcal{L}_m(H)}$, then for all $\forall t \in P^f(s)$, $\sigma \in S(t)$ if $\mathcal{L}_m(S\phi G) = \mathcal{L}_m(H)$ and $\mathcal{L}(S\phi G) = \overline{\mathcal{L}_m(H)}$.

Note that the controllability condition in Theorem 4 is also part of the necessary and sufficient conditions for controller existence with ideal sensors as discussed in Theorem 1. Controllability and $\mathcal{L}_m(G)$-closure can be tested in polynomial time using standard methods. Therefore, because of Theorem 3, controller existence with faulty sensors can then also be tested in polynomial time. Importantly, because Theorem 4 is a constructive proof, a method to synthesize controllers with faulty sensors is therefore known.

An additional benefit of the $\vec{G}$ and $\vec{H}$ constructions given above is that $\mathcal{L}_m(H)$ is controllable with respect to $\mathcal{L}(G)$ and Σ_{uc} if and only if $\mathcal{L}_m(\vec{H})$ is controllable with respect to $\mathcal{L}(\vec{G})$ and $\vec{\Sigma}_{uc}$ where $\vec{\Sigma}_{uc} = \Sigma^{\vec{G}} \setminus \vec{\Sigma}_c$. This is demonstrated in the following theorem.

Theorem 5: Suppose G, H and Σ_{uc} are given and $\vec{G}$, $\vec{H}$ and $\vec{\Sigma}_{uc}$ are constructed from them. Then, $\mathcal{L}_m(H)$ is controllable with respect to $\mathcal{L}(G)$ and Σ_{uc} if and only if $\mathcal{L}_m(\vec{H})$ is controllable with respect to $\mathcal{L}(\vec{G})$, and $\vec{\Sigma}_{uc}$.

Theorem 6: Suppose G and H are given and $\vec{G}$ and $\vec{H}$ are constructed from them. Then, $\mathcal{L}_m(H)$ is $\mathcal{L}_m(G)$-closed if and only if $\mathcal{L}_m(\vec{H})$ is $\mathcal{L}_m(\vec{G})$-closed.

The following corollary of Theorems 3, 4, 5 and 5 can now be shown which demonstrates that the standard perfect-sensor control methods of [3] can be used with the $\vec{G}$ and $\vec{H}$ to test controller existence in the faulty-sensor scenario introduced above.

Corollary 1: Consider G and H such that $\mathcal{L}_m(H) \subseteq \mathcal{L}(G)$, a set of controllable events Σ_c and a set of observable events Σ_o. From G, H, Σ_c and Σ_o, construct $\vec{G}$, $\vec{H}$, $\vec{\Sigma}_c$ and $\vec{\Sigma}_o$ as discussed above. There exists a nonblocking faulty sensor controller S such that $\mathcal{L}_m(S\phi G) = \mathcal{L}_m(H)$ and $\mathcal{L}(S\phi G) = \overline{\mathcal{L}_m(H)}$ if and only if there exists a nonblocking perfect sensor controller $\vec{S}$ such that $\mathcal{L}_m(\vec{S}/\vec{G}) = \mathcal{L}_m(\vec{H})$.

An additional convenience of the $\vec{G}$ and $\vec{H}$ constructions is that if a nonblocking controller $\vec{S}$ is synthesized under the assumption of perfect sensors such that $\mathcal{L}_m(\vec{S}/\vec{G}) = \mathcal{L}_m(\vec{H})$, then the same controller can be used in the faulty-sensor case to ensure that $\mathcal{L}_m(\vec{S}\phi G) = \mathcal{L}_m(H)$ and $\mathcal{L}(\vec{S}\phi G) = \overline{\mathcal{L}_m(H)}$.

Theorem 7: Consider a finite state automaton system model G, a finite state automaton specification H such that $\mathcal{L}_m(H) \subseteq \mathcal{L}_m(G)$, a set of controllable events Σ_c and a set of observable events Σ_o with sensors that may fail as described above. From G, H, Σ_c and Σ_o, construct $\vec{G}$, $\vec{H}$, $\vec{\Sigma}_c$ and $\vec{\Sigma}_o$ as discussed above. If a nonblocking perfect-sensor controller $\vec{S}$ is synthesized such that $\mathcal{L}_m(\vec{S}/\vec{G}) = \mathcal{L}_m(\vec{H})$ and $\mathcal{L}(\vec{S}/\vec{G}) = \overline{\mathcal{L}_m(\vec{H})}$, then $\vec{S}$ can be used in the faulty-sensor situation such that $\mathcal{L}_m(\vec{S}\phi G) = \mathcal{L}_m(H)$ and $\mathcal{L}(\vec{S}\phi G) = \overline{\mathcal{L}_m(H)}$.

Theorem 7 can be used to synthesize a controller $\vec{S}$ such that if $\mathcal{L}_m(H)$ is not sensor failure observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c, then, using standard methods, one could design $\vec{S}$ using the $\vec{G}$ and $\vec{H}$ constructions such that $\mathcal{L}_m(\vec{S}/\vec{G})$ is a maximal controllable and observable sublanguage of $\mathcal{L}_m(\vec{H})$. Then, the controller $\vec{S}$ could be used in the faulty-sensor situation such that $\mathcal{L}_m(\vec{S}\phi G)$ is also maximal in a sense.

VI. Discussion

This paper discusses supervisory control situations where the controller has sensors that may fail. A version of observability, called sensor-failure observability, is introduced that is part of the necessary and sufficient conditions for controllers to exist such that a given specification is satisfied when the controller operates on a system. A polynomial time construction is given that can be used to test for the existence of these controllers and then synthesize these controllers using standard methods.

VII. Acknowledgements

The author would like to acknowledge the helpful comments and questions from George Jiroveanu of the University of Ghent.

References

[1] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, Boston, MA, 1999.

[2] R.M. Jensen. DES controller synthesis and fault tolerant control: A survey of recent advances. Technical Report TR-2003-40, The IT University of Copenhagen, 2003.

[3] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44:173–198, 1988.

[4] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal of Control Optimization*, 25(1):206–230, 1987.

[5] K. Rohloff. Sensor failure tolerant supervisory control. Technical report, Coordinated Science Laboratory, The University of Illinois at Urbana-Champaign, Urbana, IL, USA, 2005.

[6] J. Tsitsiklis. On the control of discrete-event dynamical systems. *Mathematics of Control, Signals and Systems*, 2:95–107, 1989.